Tree Sort Algorithm

Joy Kim | AP CSA P5

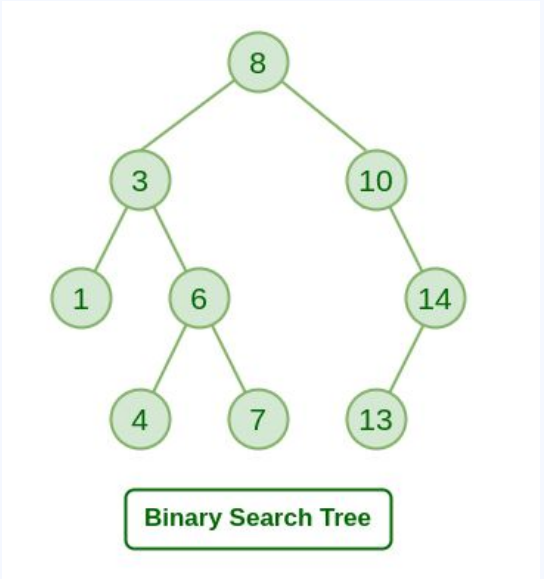
Binary Search Trees

Knowledge basing **Tree Sort**

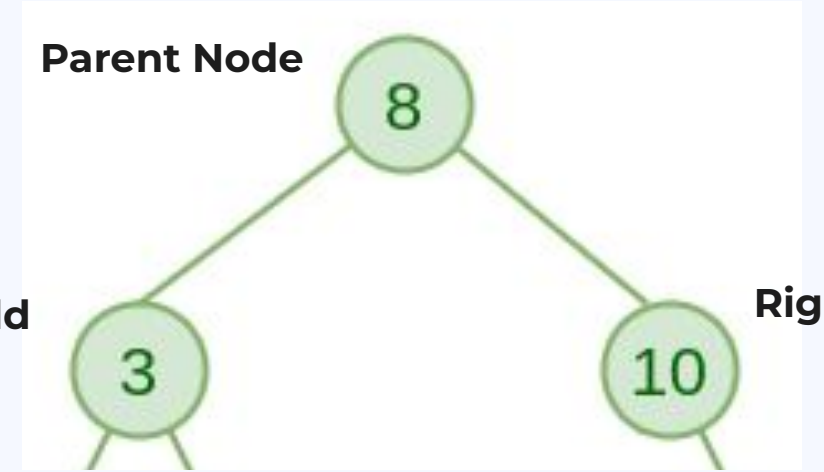
- Invented by P.F. Windley, Andrew Booth, Andrew Colin, and Thomas Hibbard in 1960

Terminologies

- Parent Node
- Left / Right Child



Binary Search Trees

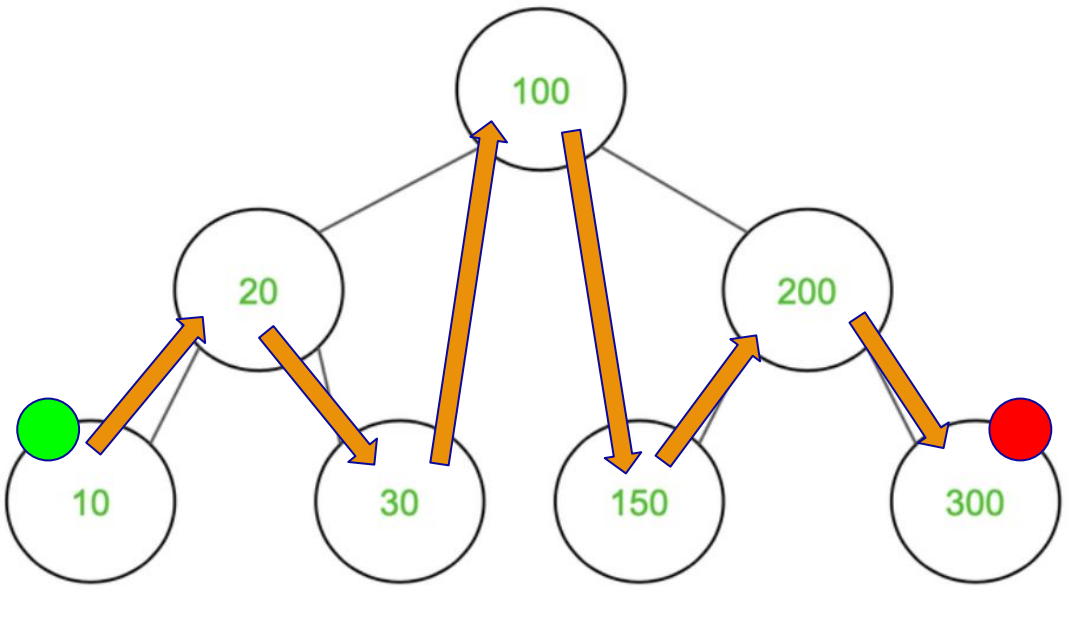


Tree Sorting

A **sorting algorithm** that takes the advantage of **BST**s to sort data

In Order Traversal

"From the **Left Subtree** to the **Right Subtree**"



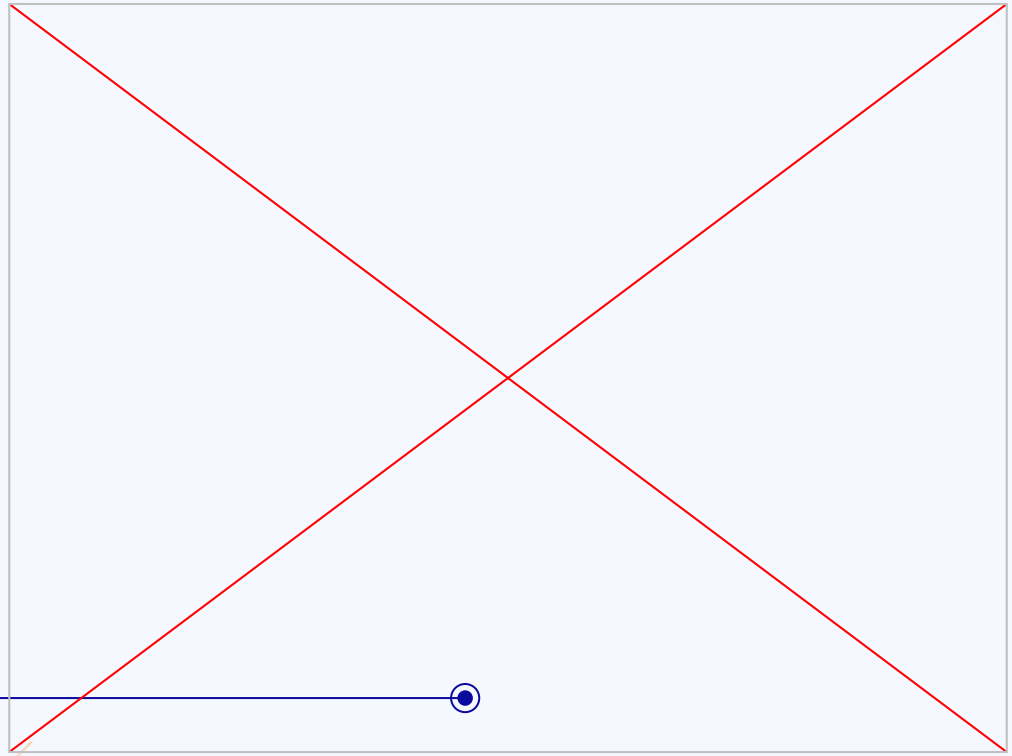
How to Tree Sort

1. **Map** the data to a **Binary Search Tree**

2. **Conduct** an **In-Order Traversal** of the BST

Code

Quick Demo

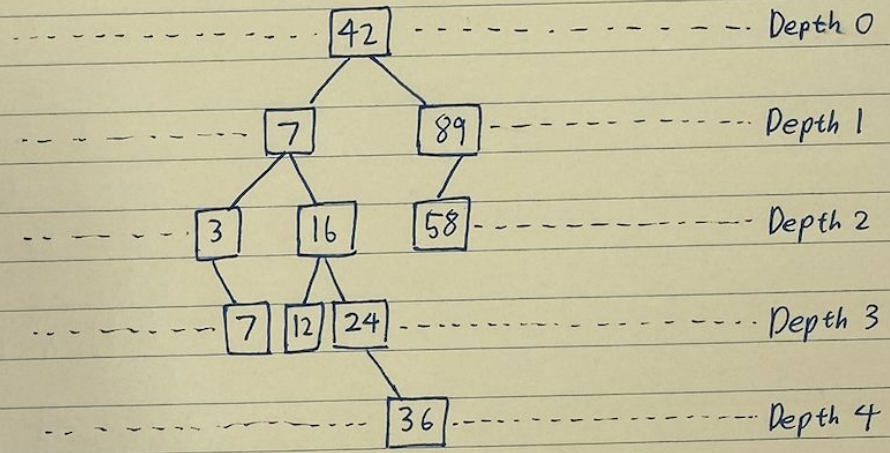


```
10
42 7 89 16 3 58 24 7 12 36
42 is put under the first node.
7 is put under the Left child of 42.
89 is put under the Right child of 42.
16 is put under the Left child of 42, and Right child of 7.
3 is put under the Left child of 42, and Left child of 7.
58 is put under the Right child of 42, and Left child of 89.
24 is put under the Left child of 42, Right child of 7, and Right child of 16.
7 is put under the Left child of 42, Left child of 7, and Right child of 3.
12 is put under the Left child of 42, Right child of 7, and Left child of 16.
36 is put under the Left child of 42, Right child of 7, Right child of 16, and Right child of 24.
3 7 7 12 16 24 36 42 58 89 %
```

INPUT: N=10

arr[] = {42, 7, 89, 16, 3, 58, 24, 7, 12, 36}

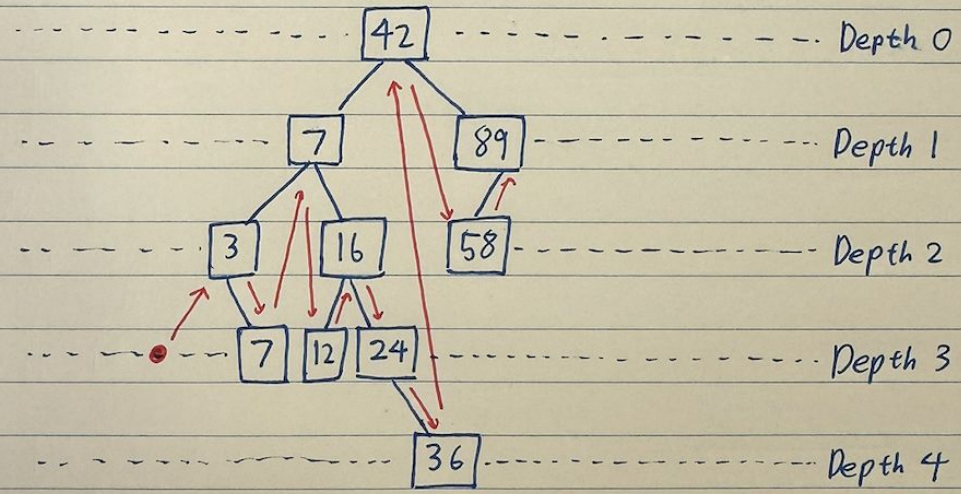
Created Binary Search Tree (BST)



INPUT: N=10

arr[] = {42, 7, 89, 16, 3, 58, 24, 7, 12, 36}

Created Binary Search Tree (BST)



OUTPUT: 3, 7, 7, 12, 16, 24, 36, 42, 58, 89

Time Complexity

O(log n) ~ O(n log n)

The idea is to do maximum of 2^x calculations to go through all n elements

⇒ x = log base 2 of n = log n ~ n log n